



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/635,830	08/09/2000	Andrew J. Layman	MS1-521US	5560
22801	7590	01/17/2006	EXAMINER	
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201			TRAN, QUOC A	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 01/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/635,830

Applicant(s)

LAYMAN ET AL.

Examiner

Quoc A. Tran

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 26 July 2005.  
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.  
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-42, 45 and 48 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.  
6) ☒ Claim(s) 1-18, 42, 45 and 48 is/are rejected.  
7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.  
8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.  
10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)  
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 12/05/2005  
4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_  
5) ☐ Notice of Informal Patent Application (PTO-152)  
6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. This action is responsive to Amendment: filed 07/26/2005, with acknowledgement of original filed 08/09/2000.
2. Claims 1-18, 42, 45 and 48 are currently pending in this application. Applicants amended independent claims 1, 16-17, 42, 45 and cancelled claims 43-44, 46-47 and 49-50. Claims 1, 16-17, 42, 45 and 48 are independent claims.
3. Claims 1-14 and 17-18 and 22 were rejected under 35 U.S.C. 101 have been withdrawn.

### *Claim Rejections - 35 USC § 103*

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

*(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.*

**Claims 1-18, 42, 45 and 48** are rejected under 35 U.S.C. 103(a) as being unpatentable by Martin et al. (hereinafter Martin) "Professional XML", Public Release 01/2000, By Wrox Press Ltd, Birmingham, UK, in view of Martin – US006704743B1 provisional No.60/153,651 - filed 09/13/1999 (hereinafter Martin '743).

**In regard to independent claim 1**, "generating a data structure ("datastruct") representative of a data structure of a first object the datastruct element ....", as taught by Martin, at pages 74-79, chapter 3, especially bottom of page 74 (i.e....XML element types...attribute assigned to a specific element type ....),

*“a first object the datastruct element”*, as taught by Martin, at pages 111-113, chapter 4, especially middle of page 113 (i.e....<Holiday> ....). Examiner reads as first object of the datastruct;

*“first object the datastruct element having a pair of data structure tags”*, as taught by Martin, at pages 111-113, chapter 4, especially middle of page 113

(i.e....<Holiday>.....</Holiday> ....). Examiner reads as a pair of data structure tags;

*“wherein the data structure tags identify the datastruct element”*, as taught by Martin, at pages 111-113, chapter 4, especially middle of page 113 (i.e....<Holiday> ....</Holiday>). Examiner reads as data structure tags <Holiday>...</Holiday> is identify the element of the first object, which was define as “Holiday”.

*“ generating contents of the datastruct element between the datastruct the contents comprising one or more data parameter elements representative of one or more data parameter of the first object data structure”*, as taught by Martin, at pages 111-113, chapter 4, especially middle of page 113 (i.e....

<Holiday>

<Journey>.....

<from>.....</ to>

<Date>.....</Date>

</ Journey>

.....

</Holiday>

Art Unit: 2176

Examiner reads contents <Journey>..... <from>.....</ to> <Date>.....</Date> </Journey>

are data parameter elements, which generates from the first object <Holiday>,

*“each parameter element having a pair of parameter tags associated therewith, wherein each pair of parameter tags identifies a parameter element with which the pair of tags is associated, each parameter element having associated data between the pair of parameter tags”*, as taught by Martin, at pages 111-113, chapter 4, especially middle of page 113 (i.e....

<Holiday>

<Journey>.....

<from>.....</ to>

<Date>.....</Date>

</ Journey>

.....

</Holiday>.

Examiner reads contents <Journey>..... <from>.....</ to> <Date>.....</Date>

</Journey> are data parameter elements, which generates from the first object <Holiday>, and

each parameter elements having a pair of tags, that is associated with each other. For more

support to the example above, Examiner would like to include the following example as well, as

taught by Martin, at pages 109-126, chapter 4 (i.e....mapping the information... representing

object types...representing relationship....:

<book

author= “Nelson Mandela”

title="Long Walk to Freedom"

publisher= "Abacus"

isbn="0-349-10653-3" />....

Here is the same information, this time representing the properties as child elements:

<book>

<author>Nelson Mandela</author>

<title>Long Walk to Freedom</title>

<publisher>Abacus</publisher>

<isbn>0-349-10653-3</isbn>

</book>....)

Martin does not explicitly teach, **the datastruct element being** representative of a data structure of a first object, however (Martin '743 at col. 43, line 10 through col. 45, line 55, also see Fig. 31-33, discloses a GUI controls using form that includes parameters, UI elements, user interface model (see item 532, 534, 536, 538, 540 and 540). Using the broadest reasonable interpretation to the claim limitation, wherein datastruct element being generated (e.g. GUI, UI elements...) not just the content of the element, to a person of ordinary skill in the art at the time the invention was made.

Also Martin does not explicitly teach, **embodied in one or more computer readable media**, however (as taught by Martin '743, at col. 10, lines 30-60, discloses "computer programs", or simply "programs" that is fully functioning computers and computer systems, those skilled in the art will appreciate that the various embodiments of the invention are capable of being distributed as a program product in a variety of forms, and that the invention applies

Art Unit: 2176

equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of signal bearing media include but are not limited to recordable type media such as volatile and non-volatile memory devices, floppy and other removable disks, hard disk drives, magnetic tape, optical disks (e.g., CD-ROM's, DVD's, etc.), among others, and transmission type media such as digital and analog communication links).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Martin's teaching, to provide a way, to include a means of embodied in one or more computer readable media and the datastruct element being representative of a data structure of a first object of Martin '743 teaching. One of the ordinary skill in the art would have been motivated to modify this combination to permit "computer programs", or simply "programs" that is fully functioning computers and computer systems, are capable of being distributed as a program product in a variety of forms (e.g. media) (as taught by Martin '743 at col. 9, line 60 through col. 10 line 65) and to provide solution of the limitation of restricting of the inheritance of fields and other object-oriented entity in application that rely on underlying databases and persistent storage of object0oriented entities (as taught by Martin '743, at col. 3, lines 50-67).

**In regard to dependent claim 2,** *" the contents further comprise at least cue object reference referencing a second object within the data structure of the first object without including the second object within the contents of the datastruct element, "*, as taught by Martin, at pages 122-124, chapter 4 (i.e....relationship in your model can be represented by using nested elements..."contain" relationships, for example <footnote> element can be nested within a <chapter> element, and <hotel> element can be nested within a <resort> element....

```
<Itinerary>

<traveler id= "t01">

<name> Mrs Mary Higgins</ name >

</traveler>

<traveler id=" t02">

< name > Mr John Higgins</name>

</traveler>

<traveler id="t03">

< name > Rory Higgins</name>

<age> 12</age>

</ traveler>

<traveler id='t04'>

<name> Kylie Higgins</name>

<age> 9< / age>

</traveler>

.....

</itinerary>...).
```

**In regard to dependent claim 3**, Martin does not explicitly teach, “*the second object is the first object*”, however (Martin ‘743, at col. 30, lines 10-25 (i.e.... FIGS. 20A—20C illustrate a working example of implicit inheritance. As shown in FIG. 20A, for example, a working storage 300 and a persistent storage 302 are illustrated as including a parent entity (“customer’ having a persistent representation 304 with a customer number ... An exemplary pair of child



Art Unit: 2176

entities (“customer #223” and “customer #224”) with persistent representations 310, 314 are also illustrated, with each including a customer number field 312, 316 that overrides the customer number field 306 of the parent entity. However, given that each child entity does not explicitly override the preferred field 308, each child entity is considered to implicitly inherit this field from the parent entity. ...).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Martin ‘743 into Martin’s teaching, to provide a way, wherein the second object is the first object. One of the ordinary skill in the art would have been motivated to modify this combination to provide solution of the limitation of restricting of the inheritance of fields and other object-oriented entity in application that rely on underlying databases and persistent storage of object-oriented entities, as taught by Martin ‘743, at col. 3, lines 50-67 (i.e. ... of the limitation of restricting of the inheritance of fields and other object-oriented entity...).

**In regard to dependent claim 4**, *“the contents comprises a data type definition for at least one data parameter element”*, as taught by Martin, at pages 121-122, chapter 4 (i.e....DTD (data type definition)...define...attributes and child element...).

**In regard to dependent claim 5**, *“the contents comprises a reference to a data type definition for at least one data parameter element.”*, as taught by Martin, at pages 121-122, chapter 4 (i.e....DTD (data type definition)...define...attributes and child element...).

**In regard to dependent claims 6-7**, incorporate substantially similar subject matter as cited in claims 1 above, and is similarly rejected along the same rationale.

**In regard to dependent claims 8-9**, incorporate substantially similar subject matter as cited in claims 1 and 4-5 above, and is similarly rejected along the same rationale.

**In regard to dependent claim 10**, *“the datastruct element and its contents are encoded using XML”*, as taught by Martin, at pages 75-76, chapter 3 (i.e.... formal DTD structure...XML...).

**In regard to dependent claim 11**, *“inserting the datastruct element into a message; and sending the message to an entity on a network”*, as taught by Martin, at pages 504-510, chapter 11, see figures at pages 506-507 (i.e ... XML-RPC (xml remote procedure calling) and SOAP (simple object access protocol) protocol...).

**In regard to dependent claim 12**, *“formatting the message for sending over a network using HTTP; sending the message to an entity on the network by using HTTP”*, as taught by Martin, at pages 507-509, chapter 11, see figure top of page 509 (i.e....XML-RPC operates over pure HTTP...).

**In regard to dependent claim 13**, *“binding the message into a HTTP request; sending the message to an entity on the network by using HTTP”*, as taught by Martin, at pages 507-509, chapter 11, see figure top of page 509 (i.e. ...XML-RPC operates over pure HTTP...).

**In regard to dependent claim 14**, incorporate substantially similar subject matter as cited in claim 1 above, and is similarly rejected along the same rationale.

**In regard to dependent claim 15**, is directed to a computer readable storage medium for performing the method of claim 1 above, and is similarly rejected along the same rationale.

**In regard to dependent claim 16**, incorporate substantially similar subject matter as cited in claims 1 and 10 above, and is similarly rejected along the same rationale.

**In regard to dependent claim 17**, incorporate substantially similar subject matter as cited in claim 1 above, and is similarly rejected along the same rationale.

**In regard to dependent claim 18**, incorporate substantially similar subject matter as cited in claim 3 above, and is similarly rejected along the same rationale.

**In regard to independent claim 42**, is directed to a computer readable storage medium for performing the method of claim 1 above, and is similarly rejected along the same rationale.

**In regard to independent claim 45**, is directed to an apparatus for performing the method of claim 1 above, and is similarly rejected along the same rationale.

**In regard to independent claim 48**, is directed to a computer for performing the method of claims 1 and 10 above, and is similarly rejected along the same rationale.

### ***Response to Argument***

5. Applicant's arguments with respect to claim 11-18, 42, 45 and 48 have been fully considered but are moot in view of the new ground(s) of rejection. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action.

As to the amended independent claims 1, 16, 17, 42 and 45 are mood in view of new rejection under 35 U.S.C. 103(a) as being unpatentable by Martin et al "Professional XML", Public Released 01/2000, By Wrox Press Ltd, Birmingham, UK and Martin – US006704743B1 – provisional No. 60/153,651 - filed 09/13/1999.

As for the non-amended claims (e.g. 2-15, 18 and 46-48), it is noted that Martin et al "Professional XML" and Martin – US006704743B1 fairly teaches and/or suggest the claims' limitations.

Art Unit: 2176

Additionally, the main thrust of the applicant's argument is Martin, in view of Martin are not teaching "datastruct element" which are being generated and not the contents of the "datastruct". Using the broadest reasonable interpretation of the claims, see Martin '743 at col. 43, line 10 through col. 45, line 55, also see Fig. 31-33, discloses a GUI controls using form that includes parameters, UI elements, user interface model (see item 532, 534, 536, 538, 540 and 540). Using the broadest reasonable interpretation to the claim limitation, wherein datastruct element being generated (e.g. GUI, UI elements...) not just the content of the element, to a person of ordinary skill in the art at the time the invention was made.

Further more, if applicant desires to claim the benefit of a prior-filed application under 35 U.S.C. 119 (e), a specific reference to the prior-filed application in compliance with 37 CFR 1.78(a) must be included in the first sentence(s) of the specification following the title or in an application data sheet. **For benefit claims under 35 U.S.C. 120, 121 or 365(c), the reference must include the relationship (i.e., continuation, divisional, or continuation-in-part) of the applications.**

If the instant application is a utility or plant application filed under 35 U.S.C. 111(a) on or after November 29, 2000, the specific reference must be submitted during the pendency of the application and within the later of four months from the actual filing date of the application or sixteen months from the filing date of the prior application. If the application is a utility or plant application which entered the national stage from an international application filed on or after November 29, 2000, after compliance with 35 U.S.C. 371, the specific reference must be submitted during the pendency of the application and within the later of four months from the date on which the national stage commenced under 35 U.S.C. 371(b) or (f) or sixteen months

Art Unit: 2176

from the filing date of the prior application. See 37 CFR 1.78(a)(2)(ii) and (a)(5)(ii). This time period is not extendable and a failure to submit the reference required by 35 U.S.C. 119(e) and/or 120, where applicable, within this time period is considered a waiver of any benefit of such prior application(s) under 35 U.S.C. 119(e), 120, 121 and 365(c). A benefit claim filed after the required time period may be accepted if it is accompanied by a grantable petition to accept an unintentionally delayed benefit claim under 35 U.S.C. 119(e), 120, 121 and 365(c). The petition must be accompanied by (1) the reference required by 35 U.S.C. 120 or 119(e) and 37 CFR 1.78(a)(2) or (a)(5) to the prior application (unless previously submitted), (2) a surcharge under 37 CFR 1.17(t), and (3) a statement that the entire delay between the date the claim was due under 37 CFR 1.78(a)(2) or (a)(5) and the date the claim was filed was unintentional. The Director may require additional information where there is a question whether the delay was unintentional. The petition should be addressed to: Mail Stop Petition, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

If the reference to the prior application was previously submitted within the time period set forth in 37 CFR 1.78(a), but not in the first sentence(s) of the specification or an application data sheet (ADS) as required by 37 CFR 1.78(a) (e.g., if the reference was submitted in an oath or declaration or the application transmittal letter), and the information concerning the benefit claim was recognized by the Office as shown by its inclusion on the first filing receipt, the petition under 37 CFR 1.78(a) and the surcharge under 37 CFR 1.17(t) are not required. Applicant is still required to submit the reference in compliance with 37 CFR 1.78(a) by filing an amendment to the first sentence(s) of the specification or an ADS. See MPEP § 201.11.

In addition, it is noted that the features upon which applicant relies upon the Provisional Patent Application (i.e., XML Object Access Protocol, such as XOAP defines an “XML-RPC” (XML Remote Procedure Call) protocol for client-server interaction across a network using: HTTP as the base transport and XML documents for encoding of invocation requests and responses XOAP is both low-entry and high-function, capable of use for both simple stateless remote procedure calls as well as rich object systems. XOAP works with today's deployed World Wide Web, and provides extensibility mechanisms for future enhancements. For example, XOAP supports submitting invocations using both M-POST and POST) are not recited in the rejected claim(s) (e.g., **generating a data structure (“datastruct”) element embodied in one or more computer-readable media, the datastruct element being representative of a data structure of a first object, the datastruct element having a pair of data structure tags, wherein the data structure tags identify the datastruct element; generating contents of the datastruct between the datastruct element between the datastruct tags, the contents comprising one or more data parameter elements representative of one or more data parameter of the first object data structure each parameter element having a pair of parameter tags associated therewith, wherein each pair of parameter tags identifies a parameter element with which the pair of tags is associated, each parameter element having associated data between the pair of parameter tags.** Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Therefor Examiner has not been given the above any patentable weight, in regards to effective filing date of current application (08/09/2000), for at least the reason states above at this time.

In addition to the rejection under 35 U.S.C. and 103(a) as being unpatentable by Martin et al "Professional XML", Public Released 01/2000, By Wrox Press Ltd, Birmingham, UK and Martin – US006704743B1 – provisional No. 60/153,651 - filed 09/13/1999.

Examiner has presented a second independent rejection under 35 U.S.C. 103(a) as being unpatentable by Merrick et al. US 20050166209A1 continuation of 09/274,979 filed on 03/23/1999, in view of Allen – US006658625B1 - filed 04/14/1999 as set forth below.

**Please note, Examiner would like to include a second independent rejection set forth below:**

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

***(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.***

**Claims 1-18, 42, 45 and 48** are rejected under 35 U.S.C. 103(a) as being unpatentable by Merrick et al. US 20050166209A1 continuation of 09/274,979 filed on 03/23/1999 (hereinafter Merrick), in view of Allen – US006658625B1 - filed 04/14/1999 (hereinafter Allen).

**In regard to independent claim 1, generating a data structure ("datastruct") element** (Merrick at page 11 paragraph [0037] through page 5 paragraph [0043], discloses an improved remote procedure call mechanism (RPC), wherein a generic XML-based encoding uses element type names and attribute names (the labels) to describe the data structures and the data types to which the data items conform to build the data structures that a generic message depicts) Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein a data structure element would have been an obvious variant of the data structures and the data types to which the data items conform to build the data structures that a generic message depicts,

**the datastruct element being representative of a data structure of a first object,** (Merrick at page 11 paragraph [0037] through page 5 paragraph [0043], discloses an improved remote procedure call mechanism (RPC), wherein objects are achieved, according to a first aspect of the present invention, by a method and system for invoking a service at a first machine from a second machine, in which the message encoding is self-describing, the message encoding is a generic XML-based encoding uses element type names and attribute names (the labels) to describe the data structures and the data types to which the data items conform to build the data structures that a generic message depicts),

Merrick does not explicitly teach, **the contents comprising one or more data parameter elements representative of one or more data parameter of the first object's data structure,** however (Allen at col. 15, line 5 through col. 27, line 35, also see Fig. 2-6, discloses a Program Call Markup Language (PCML), wherein a ProgramCallDocument object to retrieve and convert particular data elements in the data or data stream received from the server program/API and to convert and order data parameters that are sent to the server program/API, such as:

"<program>" and "</program>", which encapsulate the definition of a single program.



A <program> tag contains one <data> or <struct> tag for each parameter of the program.

The tags "<struct>" and "</struct>", which define a named structure that can be specified as an argument to a program or as a field within another named structure.

A <struct> tag contains one <data> or <struct> tag for each field in the structure.

The "<data>" tag, which defines a field within <program> or <struct>

Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein data parameter elements representative of one or more data parameter of the first object's data structure would have been an obvious variant of "<program>" and "</program>", which encapsulate the definition of a single program.

A <program> tag contains one <data> or <struct> tag for each parameter of the program.

The tags "<struct>" and "</struct>", which define a named structure that can be specified as an argument to a program or as a field within another named structure.

A <struct> tag contains one <data> or <struct> tag for each field in the structure.

The "<data>" tag, which defines a field within <program> or <struct>

to a person of ordinary skill in the art at the time the invention was made,

**the datastruct element having a pair of data structure tags, wherein the data structure tags identify the datastruct element**, however (Allen at col. 15, line 5 through col. 27, line 35, also see Fig. 2-6, discloses a Program Call Markup Language (PCML), wherein a ProgramCallDocument object to retrieve and convert particular data elements in the data or data stream received from the server program/API and to convert and order data parameters that are sent to the server program/API, such as:

"<program>" and "</program>", which encapsulate the definition of a single program.

Art Unit: 2176

A `<program>` tag contains one `<data>` or `<struct>` tag for each parameter of the program.

The tags "`<struct>`" and "`</struct>`", which define a named structure that can be specified as an argument to a program or as a field within another named structure (which define a named structure that can be specified as an argument to a program or as a field within another named structure. A `<struct>` tag contains one `<data>` or `<struct>` tag for each field in the structure. The "`<data>`" tag, which defines a field within `<program>` or `<struct>`.

A `<struct>` tag contains one `<data>` or `<struct>` tag for each field in the structure.

The "`<data>`" tag, which defines a field within `<program>` or `<struct>`, wherein

**A `<struct>` tag**

```
<struct name="name"
```

```
[ count="{ number .vertline. data-name }" ]
```

```
[ maxvrm="version-string" ]
```

```
[ minvrm="version-string" ]
```

```
[ offset="{ number .vertline. data-name }" ]
```

```
[ offsetfrom="{ number .vertline. data-name .vertline. struct-name }" ]
```

```
[ outputsize="{ number .vertline. data-name }" ]
```

```
[ usage="{ inherit .vertline. input .vertline. output .vertline. inputoutput }" ]
```

```
>
```

```
<struct>
```

wherein

<u>Attribute</u>	<u>Value</u>	<u>Description</u>
------------------	--------------	--------------------

Art Unit: 2176

name= "name" Specifies the name of this <struct> element

.....

Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein datastruct element having a pair of data structure tags would have been an obvious variant of the tags "<struct>" and "</struct>" which define a named structure that can be specified as an argument to a program or as a field within another named structure. A <struct> tag contains one <data> or <struct> tag for each field in the structure. The "<data>" tag, which defines a field within <program> or <struct> to a person of ordinary skill in the art at the time the invention was made,

**generating contents of the datastruct between the datastruct element between the datastruct tags, the contents comprising one or more data parameter elements representative of one or more data parameter of the first object data structure each parameter element having a pair of parameter tags associated therewith, wherein each pair of parameter tags identifies a parameter element with which the pair of tags is associated, each parameter element having associated data between the pair of parameter tags, however (Allen at col. 15, line 5 through col. 27, line 35, also see Fig. 2-6, discloses a Program Call Markup Language (PCML), wherein a ProgramCallDocument object to retrieve and convert particular data elements in the data or data stream received from the server program/API and to convert and order data parameters that are sent to the server program/API, such as:**

"<program>" and "</program>", which encapsulate the definition of a single program.

A <program> tag contains one <data> or <struct> tag for each parameter of the program.

The tags "<struct>" and "</struct>", which define a named structure that can be specified as an argument to a program or as a field within another named structure (which define a

Art Unit: 2176

named structure that can be specified as an argument to a program or as a field within another named structure. A `<struct>` tag contains one `<data>` or `<struct>` tag for each field in the structure. The "`<data>`" tag, which defines a field within `<program>` or `<struct>`.

A `<struct>` tag contains one `<data>` or `<struct>` tag for each field in the structure.

The "`<data>`" tag which defines a field within `<program>` or `<struct>`, wherein

**A `<struct>` tag**

```
<struct name="name"
```

```
[ count="{ number .vertline. data-name }" ]
```

```
[ maxvrm="version-string" ]
```

```
[ minvrm="version-string" ]
```

```
[ offset="{ number .vertline. data-name }" ]
```

```
[ offsetfrom="{ number .vertline. data-name .vertline. struct-name }" ]
```

```
[ outputsize="{ number .vertline. data-name }" ]
```

```
[ usage="{ inherit .vertline. input .vertline. output .vertline. inputoutput }" ]
```

```
>
```

```
<struct>
```

wherein

<u>Attribute</u>	<u>Value</u>	<u>Description</u>
name=	"name"	Specifies the name of this <code>&lt;struct&gt;</code> element

.....

Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein datastruct element having a pair of data structure tags and each parameter element having associated data between the pair of parameter tags would have been an obvious variant of the tags "<struct>" and "</struct>" which define a named structure that can be specified as an argument to a program or as a field within another named structure. A <struct> tag contains one <data> or <struct> tag for each field in the structure. The "<data>" tag, which defines a field within <program> or <struct> and A <program> tag contains one <data> or <struct> tag for each parameter of the program, to a person of ordinary skill in the art at the time the invention was made,

**embodied in one or more computer-readable media**, however (Allen at col. 10, lines 10-40, discloses a Program Call Markup Language (PCML), wherein equally regardless of the particular type of signal bearing media to actually carry out the distribution. Examples of signal bearing media include recordable type media such as floppy disks (e.g., disk 180) and CD ROMS, and transmission type media such as digital and analog communication links, including wireless communication links).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Merrick's teaching, wherein generating a data structure ("datastruct") element the datastruct element being representative of a data structure of a first object, to includes a means of generating contents of the datastruct between the datastruct element between the datastruct tags, such that data parameter elements representative of one or more data parameter of the first object data structure each parameter element having a pair of parameter tags associated therewith, wherein each pair of parameter tags identifies a

Art Unit: 2176

parameter element with which the pair of tags is associated, each parameter element having associated data between the pair of parameter tags, and embodied in one or more computer-readable media of Allen's teaching. One of the ordinary skill in the art would have been motivated to modify this combination for the advantages of having an RPC framework whose simplicity allows two programs to engage in RPC communications over a network without requiring software from a single vendor to be installed on both of the computers that are running the programs. Software from different vendors should be compatible by virtue of using the nearly ubiquitous HTTP protocol stack to transmit messages whose encodings use a well-defined and well-accepted text-based syntax (as taught by Merrick at page 4 paragraph [0036]).

**In regard to independent claim 16**, incorporate substantially similar subject matter as cited in claim 1 above, and further view of the following, and is similarly rejected along the same rationale,

**Encoded in XML**, (Merrick at page 11 paragraph [0037] through page 5 paragraph [0043], discloses an improved remote procedure call mechanism (RPC), wherein objects are achieved, according to a first aspect of the present invention, by a method and system for invoking a service at a first machine from a second machine, in which the message encoding is self-describing, the message encoding is a generic XML-based encoding uses element type names and attribute names (the labels) to describe the data structures and the data types to which the data items conform to build the data structures that a generic message depicts).

**In regard to independent claims 17 and 42**, incorporate substantially similar subject matter as cited in claim 1 above, and is similarly rejected along the same rationale.

**In regard to independent claim 45**, incorporate substantially similar subject matter as cited in claim 1 above, and further view of the following, and is similarly rejected along the same rationale,

**an object serializer**, (Merrick at page 11 paragraph [0037] through page 5 paragraph [0043], discloses an improved remote procedure call mechanism (RPC), wherein objects are achieved, parsed , saved according to a first aspect of the present invention, by a method and system for invoking a service at a first machine from a second machine, in which the message encoding is self-describing, the message encoding is a generic XML-based encoding uses element type names and attribute names (the labels) to describe the data structures and the data types to which the data items conform to build the data structures that a generic message depicts) Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein an object serializer would have been an obvious variant of Remote Procedure Call mechanism (RPC), wherein objects are achieved, parsed , saved according to a first aspect of the present invention, by a method and system for invoking a service at a first machine from a second machine, to a person of ordinary skill in the art at the time the invention was made.

**In regard to independent claim 48**, incorporate substantially similar subject matter as cited in claim 16 above, and is similarly rejected along the same rationale.

**In regard to dependent claims 2-4**, incorporate substantially similar subject matter as cited in claim 10 above, and are similarly rejected along the same rationale.

**In regard to dependent claims 5-10**, incorporate substantially similar subject matter as cited in claims 1 and 16 above, and further view of the following, and is similarly rejected along the same rationale,

**the contents comprise a reference to a data type definition for at least one data parameter element**, (Merrick at page 2 paragraphs [0010]-[0012], described two kinds of labels: type labels and semantic labels. A type label identifies the data type of each associated data item. Data types include the primitive data types that programming languages define. The primitive data types include integers, floats, longs, strings, and booleans. According to this definition, data types also include primitive data structures, such as arrays, records, and vectors (e.g. the Java Vector class). Data types provide the information that is necessary to represent the data item in memory so that a programming language may use the data item).

**In regard to dependent claim 11**, incorporate substantially similar subject matter as cited in claims 1, 16 and 42 above, and further view of the following, and is similarly rejected along the same rationale.

**In regard to dependent claims 12-13**, incorporate substantially similar subject matter as cited in claims 1 and 16 above, and further view of the following, and is similarly rejected along the same rationale,

**http** (Merrick at page 11 paragraph [0037] through page 5 paragraph [0043], discloses an improved remote procedure call mechanism (RPC) with HTTP).

**In regard to dependent claim 14-15**, incorporate substantially similar subject matter as cited in claims 1 and 5 above, and further view of the following, and is similarly rejected along the same rationale,

**<parameter\_label> <parameter\_data> </parameter\_label>**

**the <parameter\_label> being one of the pair of parameter tags, the <parameter\_label> being the other of the pair of parameter tags, and the parameter\_label**



Art Unit: 2176

**identifying the data parameter element; the parameter\_data being the data associated with the parameter element identified by the parameter\_label**, however (Allen at col. 15, line 5 through col. 27, line 35, also see Fig. 2-6, discloses a Program Call Markup Language (PCML), wherein a ProgramCallDocument object to retrieve and convert particular data elements in the data or data stream received from the server program/API and to convert and order data parameters that are sent to the server program/API, such as:

"<program>" and "</program>", which encapsulate the definition of a single program.

A <program> tag contains one <data> or <struct> tag for each parameter of the program.

The tags "<struct>" and "</struct>", which define a named structure that can be specified as an argument to a program or as a field within another named structure (which define a named structure that can be specified as an argument to a program or as a field within another named structure. A <struct> tag contains one <data> or <struct> tag for each field in the structure. The "<data>" tag, which defines a field within <program> or <struct>.

A <struct> tag contains one <data> or <struct> tag for each field in the structure.

The "<data>" tag, which defines a field within <program> or <struct>, wherein

#### **A <struct> tag**

```
<struct name="name"
```

```
[ count="{ number .vertline. data-name }" ]
```

```
[ maxvrm="version-string" ]
```

```
[ minvrm="version-string" ]
```

```
[ offset="{ number .vertline. data-name }" ]
```

Art Unit: 2176

```
[ offsetfrom="{ number .vertline. data-name .vertline. struct-name }" ]
[ outputsiz="{ number .vertline. data-name }" ]
[ usage="{ inherit .vertline. input .vertline. output .vertline. inputoutput }" ]
>
<struct>
```

wherein

<u>Attribute</u>	<u>Value</u>	<u>Description</u>
name=	"name"	Specifies the name of this <struct> element

Examiner read the above in the broadest reasonable interpretation to the claim limitation, wherein **<parameter\_label> <parameter\_data> </parameter\_label>** would have been an obvious variant of the tags "<struct>" and "</struct>" which define a named structure that can be specified as an argument to a program or as a field within another named structure. A <struct> tag contains one <data> or <struct> tag for each field in the structure. The "<data>" tag, which defines a field within <program> or <struct> and a <program> tag contains one <data> or <struct> tag for each parameter of the program, to a person of ordinary skill in the art at the time the invention was made,

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Merrick's teaching, wherein generating a data structure ("datastruct") element the datastruct element being representative of a data structure of a first object, to includes a means of generating contents of the datastruct between the datastruct element between the datastruct tags, such that data parameter elements representative of one

Art Unit: 2176

or more data parameter of the first object data structure each parameter element having a pair of parameter tags associated therewith, wherein each pair of parameter tags identifies a parameter element with which the pair of tags is associated, each parameter element having associated data between the pair of parameter tags, and embodied in one or more computer-readable media of Allen's teaching. One of the ordinary skill in the art would have been motivated to modify this combination for the advantages of having an RPC framework whose simplicity allows two programs to engage in RPC communications over a network without requiring software from a single vendor to be installed on both of the computers that are running the programs. Software from different vendors should be compatible by virtue of using the nearly ubiquitous HTTP protocol stack to transmit messages whose encodings use a well-defined and well-accepted text-based syntax (as taught by Merrick at page 4 paragraph [0036]).

**In regard to independent claim 18**, incorporate substantially similar subject matter as cited in claim 1 above, and is similarly rejected along the same rationale.

### ***Conclusion***

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is (571) 272-4103. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Herndon R. Heather can be reached on (571) -272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2176

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

---

*Quoc A, Tran*  
*Patent Examiner*  
*Technology Center 2176*  
*December 7, 2005*

*William L Bashore*  
**WILLIAM BASHORE**  
**PRIMARY EXAMINER**  
*12/7/2005*